

REMARKS/ARGUMENTS

Claims 1-24 are pending in the present application. Claims 1, 4, 12, 15, 18, 21, and 24 are amended. Claims 1, 12, 18, and 24 are amended to clarify that a snapshot is updated in response to a request to modify a data block and before the data block is modified. Support for the amendments to claims 1, 4, 12, 15, 18, 21, and 24 may be located at least on page 11, lines 1-20; and Figures 9A and 9B (process for updating a snapshot). Reconsideration of the claims is respectfully requested.

I. Interview Summary

Applicants thank Examiner Linh Black for the courtesies extended to Applicants' representatives during the October 11, 2006 telephone interview. During the interview, Applicants' representatives discussed the distinctions between the claims and the cited *McGovern et al.* and *Edwards* references. The substance of the telephone interview is included in the following remarks.

II. 35 U.S.C. § 102, Alleged Anticipation Based on McGovern

The Office Action rejects claims 1-2, 7, 12-13, 18-19, and 24 under 35 U.S.C. §102(e) as being anticipated by *McGovern et al.* (US 2005/0097260), hereinafter referred to as *McGovern*. This rejection is respectfully traversed.

With respect to independent claims 1, 12, 18, and 24, the Office Action states:

As per claim 1, *McGovern et al.* teach detecting a request to modify a data block in the file system; responsive to detecting the request, writing metadata describing the data block in the file system into a snapshot image – pars. 0020, 0050, 0059, 0068, 0078, 0104. copying data for the data block in the file system to the snapshot image – pars. 0046, 0058, 0062-0063. and modifying the data block in the file system after copying of the data in the data block to the snapshot image has occurred, wherein the snapshot image may be used to return the file system to a state prior to modifying the data block in the file system – pars. 0078, 0114. ...

Claims 12-24 are rejected based on the same rationale as claims 1-9.

Office Action dated July 18, 2006, pages 2-3 and 5.

As amended, claim 1, which is representative of the other rejected independent claims 12, 18, and 24 with regard to similarly recited subject matter, reads as follows:

1. A method in a data processing system for managing data in a file system, the method comprising:
detecting a request to modify a data block in the file system;
responsive to detecting the request;

writing metadata describing the data block in the file system into a snapshot image, wherein the snapshot image is updated to maintain a consistent block-level image of the file system from a point-in-time when the snapshot was created; and copying data for the data block in the file system to the snapshot image to further update the snapshot image; and
modifying the data block in the file system after copying of the data in the data block to the snapshot image has occurred, wherein the snapshot image is usable to return the file system to a state prior to modifying the data block in the file system.

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). Applicants respectfully submit that *McGovern* does not identically show every element of the claimed invention arranged as they are in the claims. Specifically, *McGovern* does not teach or suggest “responsive to detecting the request: writing metadata describing the data block in the file system into a snapshot image, wherein the snapshot image is updated to maintain a consistent block-level image of the file system from a point-in-time when the snapshot was created; and copying data for the data block in the file system to the snapshot image to further update the snapshot image,” as recited in claims 1, 12, 18, and 24.

McGovern is directed to providing a specified retention date within a data set that is locked against deletion or modification within a WORM storage implementation. This retention date scheme does not utilize any proprietary application program interfaces (APIs) or protocols, but rather, employs native functionality within conventional file (or other data containers, data sets or block-based logical unit numbers) properties available in commonly used operating systems. In an illustrative embodiment, the retention date/time is calculated by querying the file's last-modified time prior to commit, adding the retention period to this value and thereby deriving a retention date after which the file can be released from WORM. Prior to commit, the computed retention date is stored in the file's “last access time” property/attribute field, or another metadata field that remains permanently associated with the file and that, in being used for retention date, does not interfere with file management in a WORM state. Since this field is not utilized in a WORM context, it can be adapted to store this date. Once stored, the retention date in this field is locked against modification. Where extension (never reduction) of a retention period is desired, the last access time field is updated, wherein the new retention period is added to the existing last access time value to derive a new, later retention date for the file. Upon expiry of the retention date, the system allows deletion of the expired WORM file/data set. *McGovern* does not teach

or suggest that responsive to detecting the request (to modify a data block in the file system): writing metadata describing the data block in the file system into a snapshot image, wherein the snapshot image is updated to maintain a consistent block-level image of the file system from a point-in-time when the snapshot was created; and copying data for the data block in the file system to the snapshot image to further update the snapshot image, as recited in claims 1, 12, 18, and 24.

The Office Action refers to the following portions of *McGovern* in the rejection of independent claims 1, 12, 18, and 24:

[0058] One common form of update involves the use of a "snapshot" process in which the active file system at the storage site, consisting of inodes and blocks, is captured and the "snapshot" is transmitted as a whole, over a network (such as the well-known Internet) to the remote storage site. Generally, a snapshot is an image (typically read-only) of a file system at a point in time, which is stored on the same primary storage device as is the active file system and is accessible by users of the active file system. By "active file system" it is meant the file system to which current input/output operations are being directed. The primary storage device, e.g., a set of disks, stores the active file system, while a secondary storage, e.g., a tape drive, may be utilized to store backups of the active file system. Once the snapshot is taken (i.e., the image captured), the active file system is reestablished, leaving the snapshotted version in place for possible disaster recovery. Each time a snapshot is taken, the old active file system becomes the new snapshot, and the new active file system carries on, recording any new changes. A set number of snapshots may be retained depending upon various time-based and other criteria.

[0059] "Snapshot" is a trademark of Network Appliance, Inc. It is used for purposes of this patent to designate a persistent consistency point (CP) image. A persistent consistency point image (PCPI) is a point-in-time representation of the storage system, and more particularly, of the active file system, stored on a storage device (e.g., on disk) or in other persistent memory and having a name or other unique identifier that distinguishes it from other PCPIs taken at other points in time. A PCPI can also include other information (metadata) about the active file system at the particular point in time for which the image is taken. The terms "PCPI" and "snapshot" shall be used interchangeably through out this patent without derogation of Network Appliance's trademark rights.

McGovern, paragraphs [0058] – [0059].

These portions of *McGovern* disclose that a snapshot is an image of a file system at a point in time, which may be used for disaster recovery. Each time a snapshot is taken, the old active file system becomes the new snapshot. *McGovern* does not teach or suggest that responsive to detecting the request (to modify a data block in the file system): writing metadata describing the data block in the file system into a snapshot image, wherein the snapshot image is updated to maintain a consistent block-level image of the file system from a point-in-time when the snapshot was created; and copying data for the data block in the file system to the snapshot image to further update the snapshot image, as recited in claims 1, 12, 18, and 24.

In addition, the Office Action refers to the following portions of *McGovern* in the rejection of independent claims 1, 12, 18, and 24:

[0062] Also, overlying the file system layer 250 is a specialized asynchronous volume and sub-volume (or "qtree") snapshot mirroring (or replication) application 290. This application is responsible for the generation of the mirrors at a remote destination storage volume based upon changes in snapshots at the source. The snapshot mirroring application 290 operates outside of the storage access request path 270, as shown by the direct links 292 and 294 to the TCP/IP layers 215, 210 and the file system snapshot mechanism (280). This application enables "asynchronous" mirroring of changes in the respective sub-volume. That is mirroring is written incrementally (and not in real-time with respect to changes occurring on the source sub-volume) to a destination store that can be remote and linked by a network connection. A discussion of asynchronous mirroring at a volume and sub-volume (or q-tree) level is found in U.S. patent application Ser. No. 10/100,967, entitled SYSTEM AND METHOD FOR DETERMINING CHANGES IN TWO SNAPSHOTS AND FOR TRANSMITTING CHANGES TO A DESTINATION SNAPSHOT by Michael L. Federwisch, et al., which is hereby incorporated by reference.

[0063] Likewise a synchronous volume snapshot mirroring application 298 acting on the RAID layer is provided. This application provides synchronous (real-time) mirroring to a mirror store in response to a mirror command initiated by an administrator. This mirroring creates a point-in-time image of the source that copies it as it existed at the time the mirror command is acted upon. Approaches to volume-based remote mirroring of snapshots are described in detail in commonly owned U.S. patent application Ser. No. 09/127,497, entitled FILE SYSTEM IMAGE TRANSFER by Steven Kleiman, et al. and U.S. patent application Ser. No. 09/426,409, entitled FILE SYSTEM IMAGE TRANSFER BETWEEN DISSIMILAR FILE SYSTEMS by Steven Kleiman, et al., both of which patents are expressly incorporated herein by reference.

McGovern, paragraphs [0062] – [0063].

These portions of *McGovern* disclose that snapshot asynchronous mirroring is not written in real-time with respect to changes occurring on the source sub-volume. Snapshot synchronous (real-time) mirroring occurs in response to a mirror command initiated by an administrator. *McGovern* does not teach or suggest that responsive to detecting the request (to modify a data block in the file system): writing metadata describing the data block in the file system into a snapshot image, wherein the snapshot image is updated to maintain a consistent block-level image of the file system from a point-in-time when the snapshot was created; and copying data for the data block in the file system to the snapshot image to further update the snapshot image, as recited in claims 1, 12, 18, and 24.

In view of the above, Applicants respectfully submit that *McGovern* does not teach each and every feature of independent claims 1, 12, 18, and 24, as is required under 35 U.S.C § 102(e). In addition, *McGovern* does not teach each and every feature of dependent claims 2, 7, 13, and 19 at least by virtue of their dependency on claims 1, 12, 18, and 24, respectively. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 1-2, 7, 12-13, 18-19, and 24 under 35 U.S.C § 102(e).

In addition to being dependent on independent claim 1, claim 7 is also distinguished over the *McGovern* reference based on the specific features recited therein. *McGovern* does not teach or suggest that “the writing step comprises: writing an in-use state of snapshot map entries for a snapshot map group to the snapshot image prior to any before-image data blocks referenced by the snapshot map group being written to the snapshot image,” as recited in claim 7. To the contrary, *McGovern* does not even mention writing a state of snapshot map entries for a snapshot map group to the snapshot image.

Furthermore, *McGovern* does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. *McGovern* actually teaches away from the presently claimed invention because it teaches that a new snapshot image records any new changes to a file system and replaces a previous snapshot image as opposed to continuously updating a snapshot image to maintain a persistent snapshot and a consistent block-level image of a file system as in the presently claimed invention. Absent the examiner pointing out some teaching or incentive to implement *McGovern* and continuously update a snapshot image to maintain a persistent snapshot and a consistent block-level image of a file system, one of ordinary skill in the art would not be led to modify *McGovern* to reach the present invention when the reference is examined as a whole. Absent some teaching, suggestion, or incentive to modify *McGovern* in this manner, the presently claimed invention can be reached only through an improper use of hindsight using the applicants’ disclosure as a template to make the necessary changes to reach the claimed invention.

III. 35 U.S.C. § 103, Alleged Obviousness Based on *McGovern* and *Edwards*

The Office Action rejects claims 3-6, 8-11, 14-17, and 20-23 under 35 U.S.C. §103(a) as being unpatentable over *McGovern* and further in view of *Edwards* (US 2003/0182389). This rejection is respectfully traversed.

Since claims 3-6, 8-11, 14-17, and 20-23 depend from independent claims 1, 12, and 18, respectively, the same distinctions between *McGovern* and the invention recited in claims 1, 12, and 18 apply to dependent claims 3-6, 8-11, 14-17, and 20-23. In addition, *Edwards* does not provide for the deficiencies of *McGovern* with regard to independent claims 1, 12, and 18. *Edwards* is directed to a system and method for performing an on-line check of a file system. Various function calls are modified within a file system layer of a storage operating system so that each time the particular inode is retrieved using the modified function calls, a check is performed on the inode and associated buffer trees before returning the requested inode to the calling process. *Edwards* is cited for disclosing a summary map and a snap map. *Edwards* does not teach or suggest “responsive to detecting the request: writing metadata describing the data block in the file system into a snapshot image, wherein the snapshot image is updated to maintain a consistent block-level image of the file system from a point-in-time when the snapshot was

created; and copying data for the data block in the file system to the snapshot image to further update the snapshot image,” as recited in claims 1, 12, and 18. Thus, any alleged combination of *McGovern* with *Edwards* still would not result in the invention recited in claims 1, 12, and 18 from which claims 3-6, 8-11, 14-17, and 20-23 depend. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 3-6, 8-11, 14-17, and 20-23 under 35 U.S.C. § 103(a).

In addition to being dependent on their respective independent claims, claims 3, 8, 14, and 20 are also distinguished over the *McGovern* reference based on the specific features recited therein. Claims 3 and 8 are dependent on independent claim 1; claim 14 is dependent on independent claim 12; and claim 20 is dependent on independent claim 18. *McGovern* does not teach or suggest that the snapshot image includes a snapshot summary map, a snapshot map, and a set of segments and that the snapshot summary map identifies initialized states for snapshot map pages in the snapshot map, the snapshot map contains the snapshot map pages that identify data blocks in use in the file system, and the set of segments includes copies of data blocks from the file system, as recited in claims 3, 14, and 20. To the contrary, *McGovern* does not mention initialized states for snapshot map pages in the snapshot map. *McGovern* only discloses that a file system is checked and that an inode is marked as being checked. This marking can be accomplished by modifying a tracking file or by modifying a bit within the inode's metadata. Similarly, *McGovern* does not teach or suggest marking a summary snapshot map entry as being initialized and marking a location of the snapshot map group after writing the in-use state of data blocks for the snapshot map group to the snapshot image, as recited in claim 8.

IV. Conclusion

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: October 18, 2006

Respectfully submitted,

/Gerald H. Glanzman/
Gerald H. Glanzman
Reg. No. 25,035
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants

GHG/VJA